Created by: Aiswarya Mohan

TEST PLAN - (VWO.com)

Objective

VWO is a comprehensive platform that helps you test, optimize, and personalize your websites, apps, and features across devices and platforms. The objective is to ensure the reliability, functionality, and security of the login process. This includes verifying that users can successfully log in using valid credentials, as well as assessing the platform's response to invalid credentials or other potential error scenarios. The testing aims to identify any bugs, usability issues, or security vulnerabilities related to the login functionality and to ensure a seamless and secure user experience for accessing the VWO platform.

- React 18.2.0
- MongoDB
- MySQL
- Javascript

Scope

The features and functionalities of VWO.com will be tested such as user interface, checkout process, search functionality etc.

The types of testing that will be performed include manual testing, automation testing, performance testing and accessibility testing.

The environments in which the testing will be conducted includes different browsers, operating systems and device types.

The criteria that will be used to evaluate the success of testing will be number of defects found, time taken to complete the testing process and user satisfaction ratings.

The roles and responsibilities of the team members involved in the testing, such as the test lead, testers, and developers.

The schedule and milestones for the testing, including the start and end dates, and the planned testing activities.

The tools and equipment that will be used for testing, such as testing software, hardware, and documentation templates.

SIGN IN TO VWO PLATFORM

Email address

Password

Forgot Password?

Remember me

Sign in

Sign in using SSO

Don't have an account? Start a free trial

Inclusions

Introduction: This section would provide an overview of the test plan, including its purpose, scope, and goals.

Test Objectives: This section would outline the specific objectives of the testing, such as identifying and fixing defects, improving the user experience, or achieving a certain level of performance.

- Login

Exclusion

- Support Page
- Support Widget - ZOHO chat
- Dashboard Page
- Create new Campaign
- Editor
- Run on a Website

Test Environment

The operating systems and versions that will be used for testing such as WindowsOS, macOS and Linux

The browsers and versions that will be tested such as Google Chrome, Mozilla Firefox and Microsoft Edge.

The device types and screen sizes that will be used for testing such as desktop computers, laptops, tablets, and smartphones

The network connectivity and bandwidth that will be available for testing, such as Wi-Fi, cellular, or wired connections.

The hardware and software requirements for running the test cases, such as a specific processor, memory, or storage capacity.

The security protocols and authentication methods that will be used to access the test environment, such as passwords, tokens, or certificates.

The access permissions and roles of the team members who will be using the test environment, such as testers, developers, or stakeholders.

Name	Env URL
QA	qa.vwo.com
Pre Production	prepro.vwo.com
UAT	uat.vwo.com
Production	pro.vwo.com

- **Windows 10 – Chrome, Firefox and Edge**
- **Mac OS – Safari Browser**
- **Android Mobile OS – Chrome**
- **iPhone Mobile OS - Safari**

Defect Reporting Procedure

The criteria for identifying a defect, such as deviation from the requirements, user experience issues, or technical errors.

The steps for reporting a defect, such as using a designated template, providing detailed reproduction steps, and attaching screenshots or logs.

The process for triaging and prioritizing defects, such as assigning severity and priority levels, and assigning them to the appropriate team members for investigation and resolution.

The tools and systems that will be used for tracking and managing defects, such as a defect tracking software or a project management tool.

The roles and responsibilities of the team members involved in the defect reporting process, such as testers, developers, and the test lead.

The communication channels and frequencies for updating stakeholders on the progress and status of defects.

The metrics and metrics that will be used to measure the effectiveness of the defect reporting process, such as the number of defects found, the time taken to resolve them, and the percentage of defects that were successfully fixed.

Defect Process	POC
Front end	Suma
Back end	Dev
Devops	Joe

Tools - JIRA

Test Strategy

Step:1 Create test scenarios and test cases for various features in scope.
While developing test cases, we'll use a number of test design techniques.
1. Boundary Value Analysis
2. Equivalence Class Partitioning
3. Decision table testing
4. State transition testing
5. Use case testing

We also use our expertise in creating Test Cases by applying the below:
1. Exploratory testing
2. Error guessing

We prioritize the test cases.

Step:2 Our testing procedure when we receive a request for testing:

- First, we will conduct smoke testing in order to verify if all the important functionalities of the application is working as expected or not

- If the build is not stable, we reject the build and wait for a stable build before performing in-depth testing of the application functionalities.
- Once we receive a stable build which passes smoke testing, we perform an in-depth testing using the test cases created.
- Multiple test resources will be testing the same application on multiple supported environments simultaneously.

We then report the bugs in the bug tracking tool and send dev. management the defect found on that day in a status end of the day email.

As part of the Testing, we will perform the below types of Testing:
1. Smoke testing and Sanity testing
2. Regression testing and Retesting
3. Usability testing, functional and UI testing

We repeat test cycles until we get a quality product

Step:3 We will follow the below practices to make our testing better.

• Context Driven Testing – We will be performing Testing as per the context of the given application.

• Shift Left Testing – We will start testing from the beginning stages of the development itself, instead of waiting for the stable build.

• Exploratory Testing – Using our expertise we will perform Exploratory Testing, apart from the normal execution of the Test cases.

• End to End Flow Testing – We will test the end-to-end scenario which involve multiple functionalities to simulate the end user flows.

Test Schedule

Following is the test schedule planned for the project –
Task Time Duration

Task	Dates
Create Test Plan	
Test Cases creation	
Test Cases execution	
Summary reports submission date	

Sprints to test the application

Test Deliverables

The following are to be delivered to the client:

Deliverables	Description	Target Completion Date
Test Plan	Details on the scope of the Project, test strategy, test schedule, resource requirements, test deliverables and schedule	Date
Functional Test Cases	Test Cases created for the scope defined	Date
Defect Reports	Detailed description of the defects identified along with screenshots and steps to reproduce on a daily basis.	NA
Summary Reports	Summary Reports – Bugs by Bug#, Bugs by Functional Area and Bugs by Priority	Date

Entry and Exit criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:

Requirement Analysis

Entry Criteria:
- Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

- List of Requirements are explored and understood by the Testing team
- Doubts are cleared

Test Execution

Entry Criteria:
- Test Scenarios and Test Cases Documents are signed-off by the Client
- Application is ready for Testing

Exit Criteria:
- Test Case Reports, Defect Reports are ready

Test Closure

Entry Criteria:
- Test Case Reports, Defect Reports are ready

Exit Criteria:
- Test Summary Report

Tools

The following tools will be used:
1. JIRA bug tracking tool
2. MindMap tool
3. Snipping screenshot tool
4. Word and Excel documents

Risks and Mitigations

The following are the list of risks possible and the ways to mitigate them:

Risk: Non-Availability of a Resource

Mitigation: Backup Resource Planning

Risk: Build URL is not working

Mitigation: Resources will work on other tasks

Risk: Less time for Testing

Mitigation: Ramp up the resources based on the Client needs dynamically

Approvals

Team will send different types of documents for Client Approval like below:
- Test Plan
- Test Scenarios
- Test Cases
- Reports

Testing will only continue to the next steps once these approvals are done